

PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
Ministry of Higher Education and Scientific Research



UNIVERSITY ECHAHID HAMMA
LAKHDAR - EL OUED
FACULTY OF EXACT SCIENCES
Computer Science department



End of study memory
Presented for the Diploma of

ACADEMIC LICENSE

Domain: **Mathematics and Computer Science**

Industry: **Computer Science**

Specialty: **Information Systems**

Theme

**Credit charging
Application
for Smartphones**

Presented by: Farah Roukaya
Adjiba Ali
Lehouedj Hicham

Sustained on-September-2020 from the jury:

Miss. Chourouk Guettas MA (...) Supervisor Univ. El Oued

Mr/..... MA(...) President Univ. El Oued

Mr/..... MA(...) Reporter Univ. El Oued

Academic year 2019-2020.

Thanks:

It is with great pleasure that we reserve these few lines as a sign of gratitude and deep gratitude to all those who, from near or far, have contributed to the realization and the culmination of this work.

We sincerely thank our mentors Chourouk Guettas, for her understanding, patience, presence, assistance and valuable advice that were very useful to us in implementing this project.

We thank the members of the jury who kindly reviewed and evaluated this thesis.

Finally, we gladly fulfill a duty of gratitude and thanks to all our teachers for the quality of the teaching they have kindly provided us during our studies in order to provide us with efficient training.

Dedicate:

To our dear parents, May no dedication express what we owe them, for their benevolence, affection and support ... Treasures of kindness, generosity and tenderness, in testimony of our deep loves and our great gratitude "May God keep you". We would like to dedicate this humble work as proof of respect, gratitude and gratitude to our dear parents, our families and friends to all those we love for their encouragement, support, patience and prayers.

- Thanks.....
- Dedicate.....
- I. General introduction.
- II. CHAPTER 1.....
 - 1. Introduction:
 - 2. Image Recognition:
 - a. Definition about OCR.
 - b. OCR Advantages.
 - 3. Numbers Recognition:
 - a. Definition about number recognition.
 - b. Number recognition algorithm.
 - 4. Android applications:
 - a. Definition.
 - b. Activities.
 - c. Services.
 - d. Broadcast receivers.
 - e. Content providers.
 - f. Other information.
 - 5. Balance charging application :.
 - a. Definition.
 - b. Benefits of this application.....
 - c. Helpful features in Android applications.
 - d. How to use.
- III. CHAPTER 2
- 1. Introduction:.....
- 2. Target:.....
- 3. Application work pillars :
- a. Introduction
- b. Import the image and configure it to work
- c. Processing the image and extracting text from it
- d. Send recharge card numbers
- 4. Conclusion
- IV. CHAPTER 3:
- 1. Introduction
- 2. Create the work environment :.....
- a. Development device information.....
- b. Java development kit (JDK).....
- c. Android studio.....
- d. Android Softwre development kit (SDK).....
- e. Android virtual device (AVD).....
- 3. Create the application :.....
- a. Creat application android.....
- b. Amendment to the project.....
- i. Import library

Table of contents

ii.	Modification to a classe activitymain.xml	
iii.	Modification to a file MainActivity.java	
iv.	Modification to a file AndroidManifest.xml	
v.	Modification to a files String.xml	
vi.	Add properties drawable	
c.	Compilation of the project.....	
4.	The results :	
5.	Discussion :	
V.	General conclusion	
	References	
	Prospects	

General introduction

I. General introduction:

Evolution of electronic devices these days is faster than ever, there are new technologies daily, this makes now-days devices smaller with higher specs and reasonable cost. For example laptops tablets smartwatches and especially smartphones, these devices are enabled with different operating systems. The most popular operating systems are IOS from Apple and Android from Google.

Android is an open source software stack called an operating system that's created for a wide range of devices with different forms and factors, it's designed primarily for touch screen mobile devices, Android is built on the open Linux Kernel written by Open Handset Alliance. It works on smartphones, tablets, computers, smart-TV, smartwatches and Google Wear, Each device runs with Android has a set of applications designed specifically for it. The first Android device was HTC Dream releases in 2008, Android now is the most used OS in 2020, it takes 38.61% from users globally, and 47.53% from Algeria users.

Android applications now are much helpful, for example, you don't have to remember the world map, you just need a GPS application in your mobile. You need an application to translate a word you want to know because you can't learn all languages of the world, same goes for an application that can wake you up gently and on time, as well as one that reminds you about anything you may forget it.

In this context, and in order to carry out this end-of-studies project, a proposed application that makes it possible to overcome some problems that we sometimes (especially elders) find when we write Balance recharge card numbers. Actually, not everyone can see those numbers clearly, and not everyone is patient to write all the numbers in that card. This application will easily solve the problem since it will recognize the numbers and recharge the balance automatically according to the used carrier.

In the next chapters, we will give an overview about image recognition and Optical Character Recognition (OCR), the techniques that are used in this project, our application conception and the final chapter will be a showcase of our application and the obtained results, finishing with a general conclusion and some perspectives.

II. CHAPTER 1

1. Introduction

In this chapter we will clarify the details some key points of this topic, objectives and required work. We will give a definition of recognizing pictures and their types, OCR and then we will discuss the algorithm used by OCR, finally we will give an overview of our application.

2. Image Recognition

a. Definition

Image recognition is the ability of a computer powered camera to identify objects, places, people, logos, buildings, writing and several other variables in images through the use of algorithms and machine learning concepts. Image recognition is a part of computer vision and a process to identify and detect an object or attribute in a digital video or image. Computer vision is a broader term which includes methods of gathering, processing and analyzing data from the real world. A typical image recognition algorithm includes Optical Character Recognition (OCR), Pattern matching and, gradient matching, Face recognition, License plate matching, and Scene identification.

b. OCR Advantages

The advantages of OCR are numerous, but namely: it increases the efficiency and effectiveness of office work. The ability to instantly search through content is immensely useful, especially in an office setting that has to deal with high volume scanning or high document inflow. OCR is quick ensuring the document's content remains intact while saving time as well. Workflow is increased since employees no longer have to waste time on manual labor and can work quicker and more efficiently. Taking into Consideration that the better the image quality (size, contrast, lightning) the better the recognition result.

3. Numbers Recognition

a. Definition

Number recognition can be defined as “the ability to identify and name basic numerals”^[6]. Number recognition can actually be broken down into separate skills – matching, identifying, and naming.

b. Number recognition algorithm

Generally, OCR algorithms work as follow ^[1]:

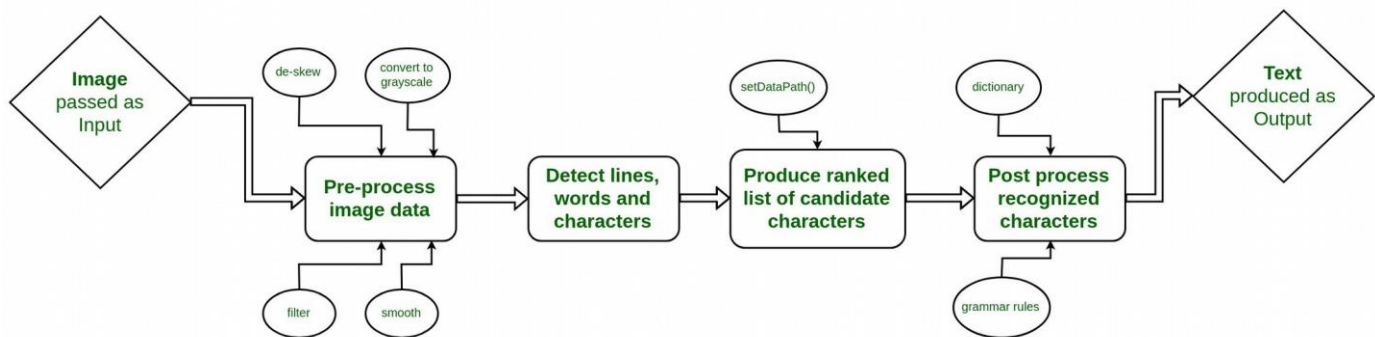


Figure 1.1. General OCR algorithm ^[2].

1. **Pre-process image data:** OCR software often “pre-process” images to boost the chances of recognition, for example:
 - a. *De-skew:* The document may not align correctly when scanned, which needs to be tilted a few degrees to create fully horizontal or vertical text lines.

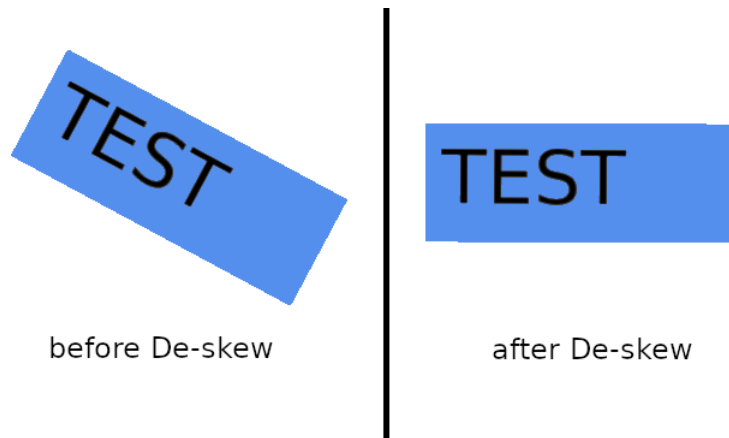


Figure 1.2. Example of De-skew pre-process step.

- b. *Gray-scale (binary image)*: Convert an image to black-and-white, called a “binary image” because there are two colors. This task is conducted as an easy and accurate way to distinguish text from the background.



Figure 1.3. Gray-scale image.

- c. *Line removal*: Cleans up non-glyph boxes and lines.
- d. *Layout analysis or “zoning”*: Identifies columns, paragraphs, captions, etc., as blocks. Particularly useful in multi-column layouts and tables.



Figure 1.4. Layout analysis.

- e. *Lines, words and characters detection*: Produce ranked list of candidate characters based on trained data set.



Figure 1.5. Characters Detection.

- f. *Character isolation or “segmentation”*: For OCR characters, various characters linked by image artifacts should be divided, single characters broken into several artifact-based pieces should be linked.



Figure 1.6. Character Isolation.

- g. *Normalization*: Normalize aspect ratio and scale.
2. **Character Extraction** : There are two main methods for extracting features in OCR:
- a. In the first method, the algorithm for feature detection defines a character by evaluating its lines and strokes.
 - b. In the second method, pattern recognition works by identifying the entire character.

We can recognize a line of text by searching for white pixel rows that have black pixels in between. Similarly, we can recognize where a character starts and finishes.

Recognition then proceeds as a two-pass process:

In the first pass, an attempt is made to recognize each word in turn. Each word that is satisfactory is passed to an adaptive classifier as training data. The adaptive classifier then gets a chance to more accurately recognize text lower down the page. Since the adaptive classifier may have learned something useful too late to make a contribution near the top of the page, a second pass is run over the page, in which words that were not recognized well enough are recognized again. A final phase resolves fuzzy spaces, and checks alternative hypotheses for the x-height to locate small-cap text ^[3].

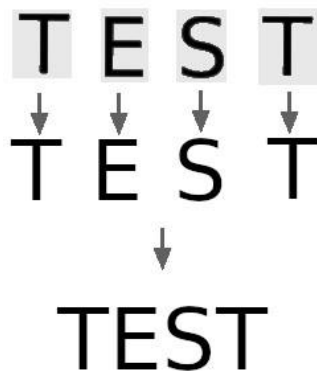


Figure 1.7. Characters Extraction and word recognition.

4. Android Applications

a. Definition

Android applications can be written using Kotlin, Java, and C++ languages. The Android SDK tools compile your code along with any data and resource files into an APK, an Android package, which is an archive file with an “.apk” suffix. One APK file contains all the contents of an Android application and is the file that Android-powered devices use to install the application ^[4].

b. Activities

An activity is the entry point for interacting with the user. It represents a single screen with a user interface. The application contains many activities work together to form a cohesive user experience, each one is independent of the others ^[4].

c. Services

A service is a general-purpose entry point for keeping an application running in the background. It is a component that runs in the background to perform long-running operations or to perform work for remote processes. A service does not provide a user interface ^[4].

d. Broadcast receivers

A broadcast receiver is a component that enables the system to deliver events to the application outside of a regular user flow, allowing the application to respond to system-wide broadcast announcements. Because broadcast receivers are another well-defined entry into the application, the system can deliver broadcasts even to applications that aren't currently running. So, for example, an application can schedule an alarm to post a notification to tell the user about an upcoming event... and by delivering that alarm to a BroadcastReceiver of the application, there is no need for the application to remain running until the alarm goes off ^[4].

e. Content providers

A content provider manages a shared set of application data that you can store in the file system, in a SQLite database, on the web, or on any other persistent storage location that your application can access. Through the content provider, other applications can query or modify the data if the content provider allows it ^[4].

f. Other information

Activities and other application components, such as services, are declared in the Android Manifest XML file ^[5].

They are first compiled to Dalvik executables to run on the Dalvik virtual machine, which is a virtual machine specially designed for mobile devices ^[5].

5. Balance charging application

1. Definition

Balance charging application is classified with “Tools” applications. Initially, the application was only for recognition of balance recharge cards, however, we added some new features to make this app more useful. Some of these features are:

- Text recognition: now you can use this application to recognize your text in any paper like any OCR application do.
- Add new Telecom Company for your Sim-card with customize entry code.
- Share the balance charge number's with any application.

2. Benefits of this application

This application has many benefits. For example:

- Help for the Blind and Visually Impaired: Screen readers can decode machine-readable text and read out the words on screen so visually impaired people can understand.
- Faster Data Entry: Generally, this application can greatly improve the effectiveness and efficiency to copy the card numbers.

6. Conclusion

In this chapter we gave a basic overview of image and number recognition, OCR algorithms and mobile applications on Android platform, alongside with a general description about the application developed in this project. In the second chapter, we will present the architecture and the design of our Application.

III. CHAPTER 2

1. Introduction

Today's artificial intelligence technology plays a very important role in our lives and in all magazines and fields, and one of these areas is pattern recognition techniques, including image recognition, such as facial recognition, object recognition, text recognition, which is what our application depends on.

2. Target

The idea of our application is to facilitate the problem of charging the SIM card without the hassle of typing a long code of numbers by taking a picture of the charging card and extracting the code and sending it to the carrier, which often ends with repeated input more than once due to an input error, which makes people vulnerable to closing the chip in the event that Error three times. Not to forget the elderly who cannot read or have low vision who often seek the help of another person to perform the operation in their place.

We hope that the application will provide help to this group that has difficulty entering the charging code for the SIM card, and it is expected that the application will reach further than that, so even young people are always looking for everything that would facilitate some work, in the shadow of the laziness that we have become suffering as a result of what it presented to us Technology from great facilities.

3. Application work pillars

a. Introduction

To enter the phone recharge code, it passes through three basic stages, according to the following scheme :

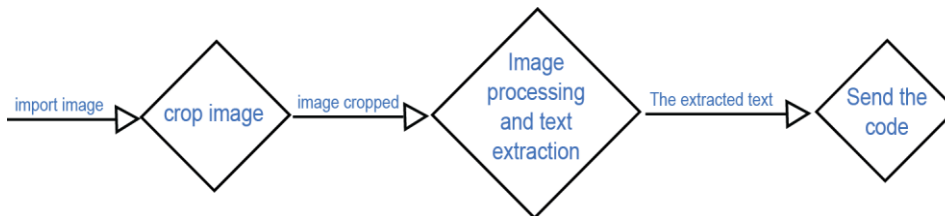


Figure 2.1. The stages of charging the SIM card.

b. Import the image and configure it to work

Importing the picture from one of the image sources on the phone (taking a picture using the camera or a picture stored in the memory).

After importing the image, a transparent box appears over it that shows the boundaries of the image, then the user changes the size of the box in proportion to the new dimensions of the

image from which the text is to be extracted.

The working principle of the image cropping algorithm is based on selecting two points from the chord, and on the basis of which the new borders of the image are determined as shown in the following diagram [6] :

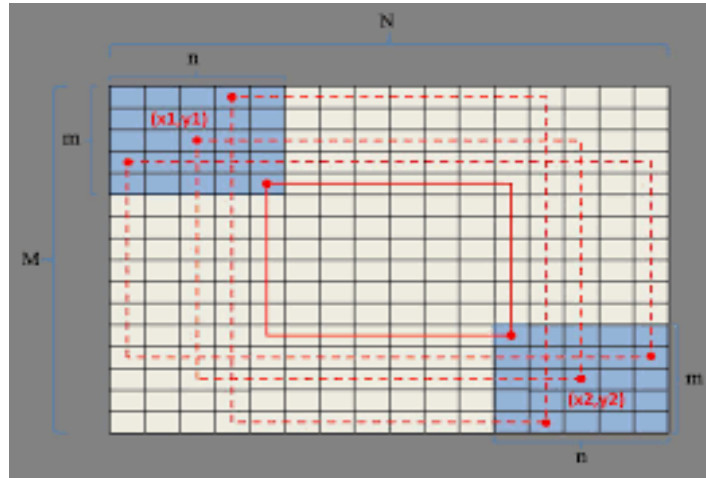


Figure 2.2. Photo Crop Matrix.

The main image is an array of pixels (N, M) dimensions [6].
 Determine the first point a (x1, y1) and the second point b (x2, y2) [6].
 The new image dimensions become after cropping (N', M') [14] :
 $N' = x2 - x1$, $M' = y2 - y1$
 On condition $1 \leq x1 < x2 \leq \text{Width}$ and $1 \leq y1 < y2 \leq \text{Height}$
 Example :

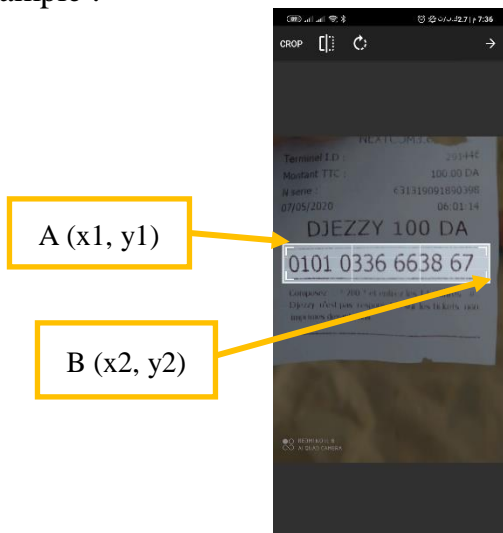


Figure 2.3. Determine the points A and B.

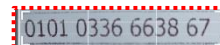


Figure 2.4. Pictures after cropping.

c. Processing the image and extracting text from it

In order to extract the text string from the image, it is initialized by a series of help analyzes [7] :

1. Gray-scale : The grayscale process equates the value of the red, green and blue layers in the RGB space so that pixels in the image are treated on the basis of one value in all layers, which is the color intensity from 0 completely to 255 completely black white according to the following equation ^[8]: $GS = (R + G + B) / 3$.

Example :

$$R = 120, G = 200, B = 50, GS = (120 + 200 + 50) / 3 = 123$$

$$\Rightarrow R = 123, G = 123, B = 123$$

2. Thresholding : In order to separate the image and distinguish the objects from the background, the local threshold value is set by calculating the average sum of the highest gray intensity and the lowest intensity in the image, and on the basis of this threshold, the image pixels are divided into two groups, a group higher than the threshold and a group less than the threshold. The value of each pixel in the group is changed to 0 for lower values and 1 for values above the threshold (or vice versa) ^[8].

Example :

The next matrix of pixels

250	200	10
10	50	150
122	100	80

$$T = (\min + \max) / 2 \Rightarrow T = (250+10) / 2 \Rightarrow \text{Thresholding} = 130$$

250	200	10
10	50	150
122	100	80

The first method : Values below the threshold are 0 and values greater than the threshold are 255

255	255	0
0	0	255
255	255	0

The second method : Values below the threshold are 255 and values greater than the threshold are 0

0	0	255
255	255	255

0	0	255
---	---	-----

1. Median Filter : In order to reduce the noise ratio in the image and obtain more accurate results, we resort to a medium filter, as it takes the average of square matrix values of individual and ordered dimensions (ascending or descending). This matrix is the pixels adjacent to the pixel to be optimized.

Example :

The pixel to optimize and the array of pixels next to it :

255	55	80
20	12	160
99	112	50

Arrange the matrix values in ascending order and specify the average values :

12	20	50	55	80	99	112	160	255
----	----	----	----	----	----	-----	-----	-----

Change the pixel value to be optimized by the average value of adjacent pixels :

255	55	80
20	80	160
99	112	50

1. Projection Approach: After separating the object from the background, we perform a horizontal and vertical projection on the object to reduce the dimensions and obtain a dimension that enables us to determine the type of the object while eliminating the zero and repeated values of the horizontal and vertical vectors Table 1 summarizes the general steps of the algorithm ^[8].

Table 1 shows the classification of numbers according to the horizontal and vertical vectors ^[8]:

H: Vector contains values those resulted from horizontal profile, after excluding the zeros and repeated values.

V: Vector contains values those resulted from vertical profile, after excluding the zeros and repeated values.

$h_i = i^{th}$ element of vector *H*, and $v_j = j^{th}$ element of vector *V*.

The vectors A and B contain additional steps for identifying the numbers two, five, and eight which are computed as follows

Store the first group of the vertical non-zeros values in matrix, that represents the binary

image, in vector A this process will continuous until get on first zero in the image and store the last group of the vertical nonzeros values in matrix that.

Represents the binary image, in vector B if the length of vector A is less than the length of vector B [8].

Properties of rows	Properties of columns	The classification
$h1 = h3 ; h1, h3 > h2$	$v1 = h3 ; v1, v3 > v2$	The number is zero
One element in vector is h1	One element in vector is v1	The number is one
$h1 = h3 ; h1, h3 > h2$	$Length(A) > Length(B)$ $v1, v3, v5 > v2, v4$	The number is two
$h2 > h1$	$v1 = v3 = v5 ; v2 = v4$	The number is three
$h3 > h1, h2$	$v2 > v1, v3$	The number is four
$h1 = h3 ; h1, h3 > h2$	$Length(A) < Length(B)$ $v1, v3, v5 > v2, v4$	The number is five
$h1 > h2, h3$	$v1 = v3 = v5 ; v2 > v4$	The number is six
$h2 > h1$	$v2 > v1$	The number is seven
$h1 = h3 ; h1, h3 > h2$	$Length(A) = Length(B)$ $v1, v3, v5 > v2, v4$	The number is eight
$h3 > h1, h2$	$v1 = v3 = v5 ; v2 < v4$	The number is nine

Table 1. Summary of general algorithm steps

Example :

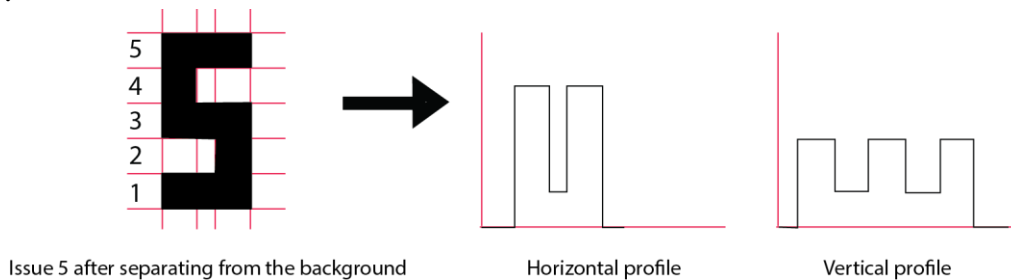


Figure 2.5. Extracting the projection from number 5

We notice in the picture above that parts 1, 3, and 5 have the same value, and parts 2 and 4 are zero values.

After deleting the zero and the repeated values, we notice in the horizontal vector that there are 3 elements with the same value and another with a small value, and in the vertical vector we notice 5 elements with the same value and 2 with the same value, but less than the three elements and we have a length of A less than the length of B, if the object is according to the table 1 is 5.

d. Send recharge card numbers

After obtaining the extracted symbol, we remove any spaces that are before or after the symbol or between the numbers.

We make sure of the length of the code so that there is no character that has been cut, the image is or is not recognized, then we add to it at the beginning the shipping service number based on the chosen chip type and the # symbol at the end, we have the shipping order ready to send and then we send it.

The following diagram shows how the transmission algorithm works :

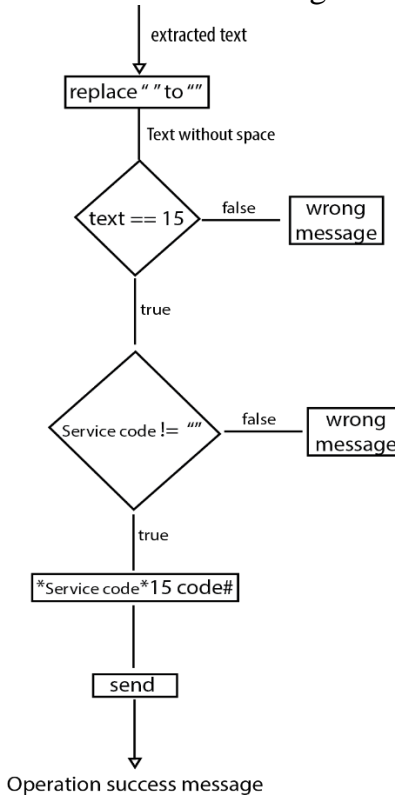


Figure 2.6. Algorithm sending recharge card code

4. Conclusion

We have seen in this chapter the methods of image processing, how to extract texts from it using artificial intelligence, and how to make contact using the phone, and now let's move on to the third chapter to follow step by step how to create a work environment and implement the application to the end

IV. CHAPTER 2

1. Introduction

To create any program that needs in the first stages to go through, starting from the identification of libraries, the interface for programming applications and the programming language that makes it easier for you to program as quickly as possible and at the highest possible efficiency and the necessary hardware to be used by programming this program and this will be discussed in this chapter.

2. Create the work environment

Any work that we want to do without a doubt needs an appropriate work environment to facilitate the work, and so is the matter with our program. This needs an appropriate work environment of hardware and software, which is what we will mention the following points:

a. Development device information :

The device information used in developing this app is summarized in the following table:

Element	Value
Device name	ASUS Notebook K95VJ
Operating System	Microsoft Windows 10 Pro
Operating System version	10.0.19041
Nucleus	X64-bit
Processor	Intel(R) Core (TM) i7-3630QM @2.40GHz Octa core
Memory	8 GB
Graphics card	NVIDIA GeForce GT 635M
Screen	18.4 pouces 16:9, 1920 x 1080 pixel
Storage	The least possible 10 GB

b. Java development kit (JDK) V1.8.0_181

The JDK allows developers to create Java programs that can be executed and run ^[9].

c. Android studio V3.6.1

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA ^[10].

d. Android Software development kit (SDK)

The Android SDK (software development kit) is a set of development tools used to develop applications

for Android platform. The Android SDK includes the following:

- Required libraries
- Debugger
- An emulator
- Relevant documentation for the Android application program interfaces (APIs)
- Sample source code
- Tutorials for the Android OS

It can be downloaded from within the Android studio

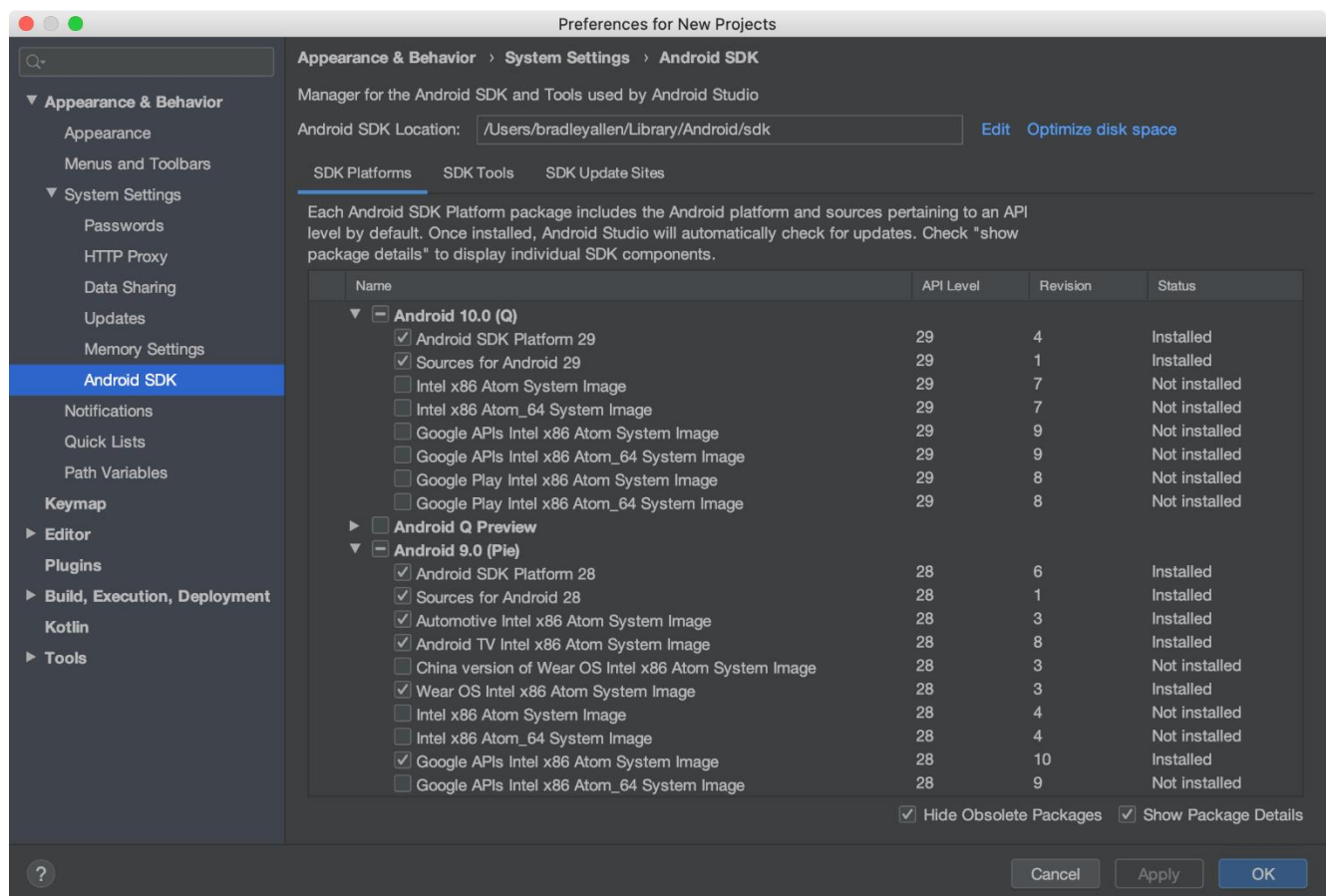


Figure 3.1. android SDK

e. Android virtual device (AVD)

An Android Virtual Device (AVD) is a configuration that defines the characteristics of an Android phone, tablet, device that you want to simulate in the Android Emulator. The AVD Manager is an interface you can launch from Android Studio that helps you create. An AVD contains a hardware profile, system image, storage area, skin, and other properties. However, the use of the emulator inside the same device requires additional resources and thus affects the performance of the ready. This is why it is preferred to use the phone as a simulator instead of the device, especially in the case of weak devices.



Despite the good features of the device used in programming our application, we preferred to use an external phone (xiaomi redmi note 8) as a simulator, and this is to be used with the phone's camera, which the application was built on.

Why Java and Android studio ?

There are many ways to build an Android application, generally mobile development has two parts cross platform and native platform. Native platform is software program that is developed for use on a particular platform or device, in this case we have Java and Kotlin for the android applications. Cross platform software is computer software that is implemented on multiple computing platforms, it's mean "write once time run everywhere"^[5], popular platforms are Flutter with dart language, and React Native, ionic, Native Script with java script language, and Xamarin with C# language. The most of android applications are written in the Java programming language and use Java core libraries.

We chose java and android studio :

- Native java code
- The final application is very small compared to the react native or ionic
- Easy to configure and simple files, unlike react native, its original files are more than 100MB
- Layout Editor

3. Create the application

a. Creat application android

In the Welcome to Android Studio window, click Start a new Android Studio project. If you have a project already opened, select **File > New > New Project**.

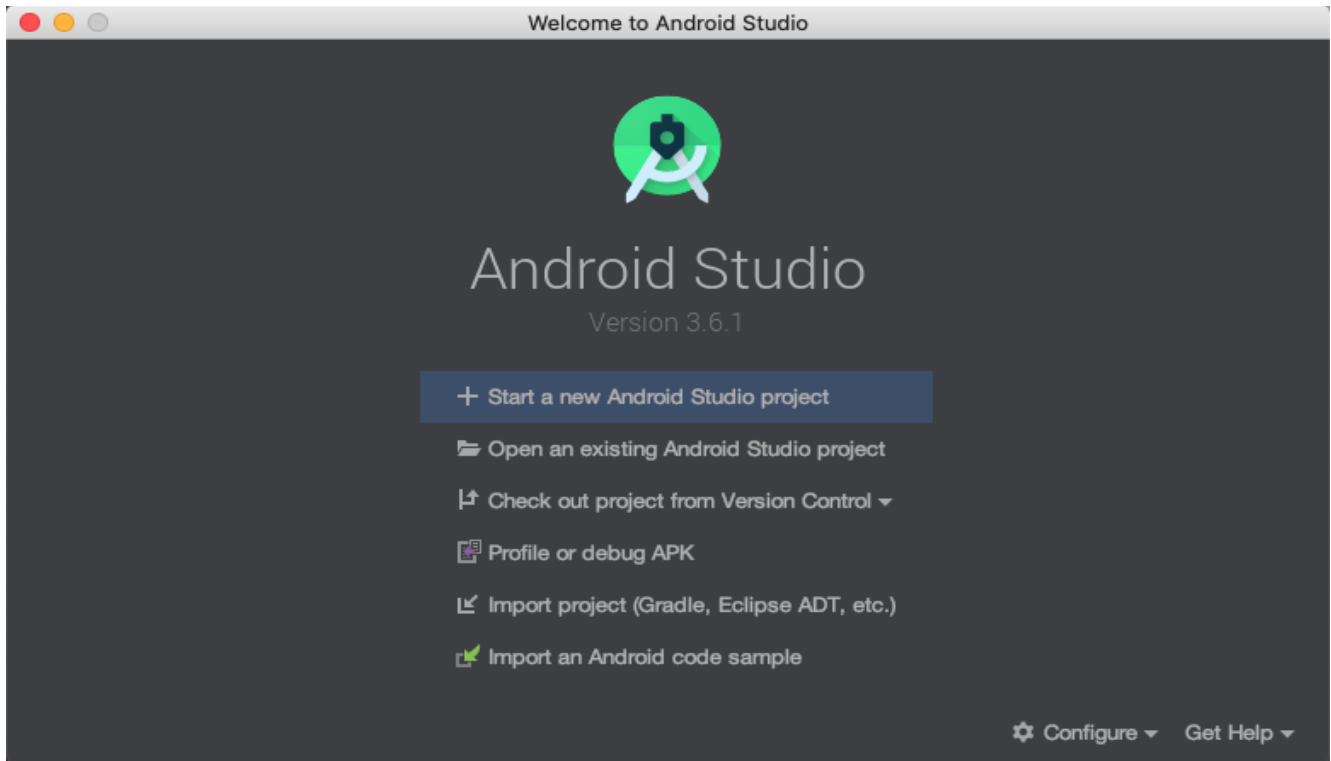


Figure 3.2. interface android studio

b. Amendment to the project

In order to accomplish this application, we need to amend the initial files and add some help libraries

i. Import library :

We add the image clipping library and the text recognition library in a file **build.gradle**

Clipping library : **'com.theartofdev.edmodo:android-image-cropper:2.6.0'**

Text recognition library : **'com.google.android.gms:play-services-vision:20.1.0'**

Chapter 3 : Implementation

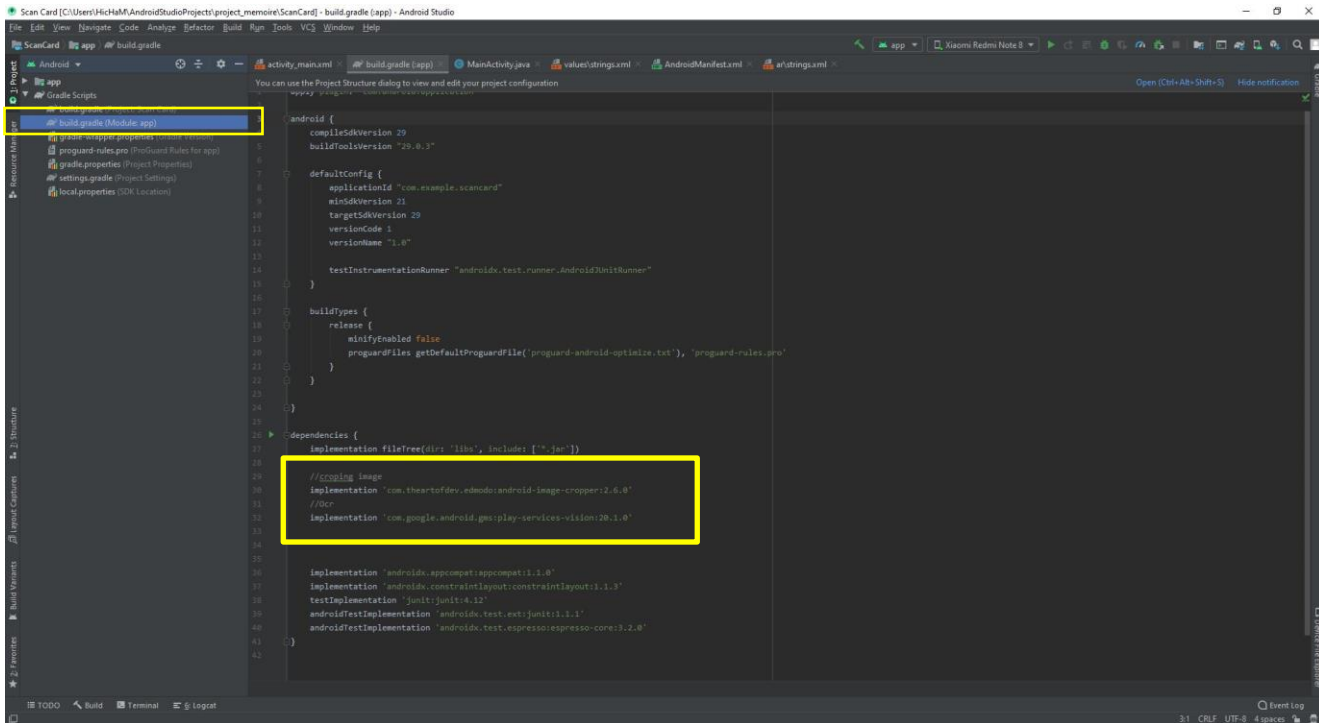


Figure 3.3. file build.gradle

ii. Modification to a class *AndroidManifest.xml*

- Add authorization lines for calls, external storage and internal storage.
- Import cropping activity.

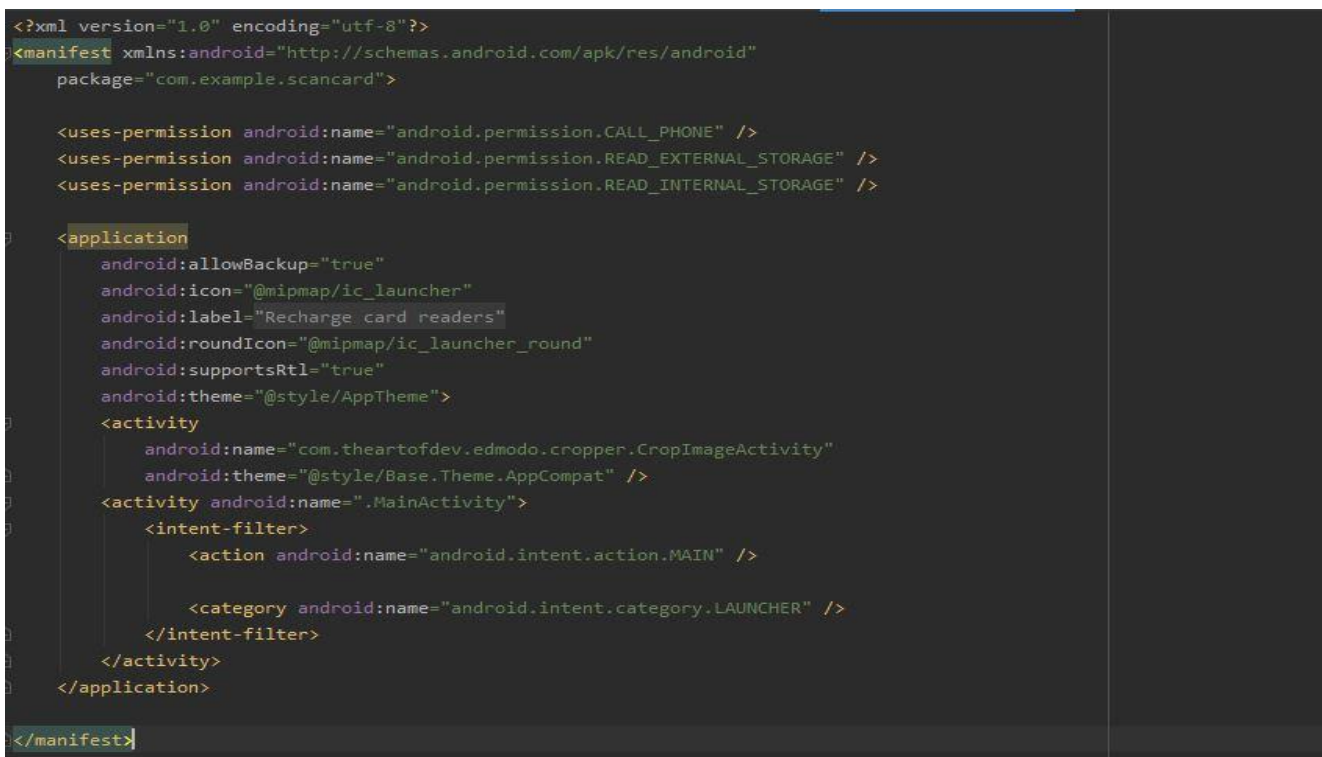


Figure 3.4. file Manifest

iii. Modification to a file *MainActivity.java*

- Import all used objects
- Define variables.
- Call a validation function that gives permission to use calls.
- Calling all UI elements.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    checkForCallPhonePermission();
    btn = findViewById(R.id.mButton);
    myImg = findViewById(R.id.mImageView);
    myText = findViewById(R.id.textView);
    myText.setText("");

    radioGroupSimCard = (RadioGroup) this.findViewById(R.id.radioGroup_SimCard);
    radioButtonMobilis = (RadioButton) this.findViewById(R.id.radioButton_Mobilis);
    radioButtonDjezzy = (RadioButton) this.findViewById(R.id.radioButton_Djezzy);
    radioButtonOoredoo = (RadioButton) this.findViewById(R.id.radioButton_Ooredoo);
}
```

Figure 3.5. onCreate function

- A function that checks the validity of using calls.
 - The function of showing the box for the question to give the permission for calls.
 - A function of showing the send button in the event of giving the call powers.
 - A function of hiding the send button in case calls are not granted.
-
- Function of importing a picture from device applications or taking a picture using camera applications
 - The function of processing the imported image, extracting the digital string, and saving it in a variable

```
public void onChooseaFile(View v) {
    CropImage.activity().start( activity, MainActivity.this);
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    if (requestCode == CropImage.CROP_IMAGE_ACTIVITY_REQUEST_CODE) {
        CropImage.ActivityResult result = CropImage.getActivityResult(data);

        if (resultCode == RESULT_OK) {
            mImageUri = result.getUri();
            myImg.setImageURI(mImageUri);

            BitmapDrawable bitmapDrawable = (BitmapDrawable) myImg.getDrawable();
            Bitmap bitmap = bitmapDrawable.getBitmap();

            TextRecognizer textRecognizer = new TextRecognizer.Builder(getApplicationContext()).build();

            if (!textRecognizer.isOperational()) {
                Toast.makeText( context: this, text: "Error", Toast.LENGTH_LONG).show();
            } else {
                Frame frame = new Frame.Builder().setBitmap(bitmap).build();
                SparseArray<TextBlock> items = textRecognizer.detect(frame);
                StringBuilder stringBuilder = new StringBuilder();

                for (int i = 0; i < items.size(); i++) {
                    TextBlock item = items.valueAt(i);
                    stringBuilder.append(item.getValue());
                    Log.d(TAG, msg: "Value Item " + i + " : " + item.getValue());
                    stringBuilder.append("\n");
                }
                myText.setText(stringBuilder.toString());
            }
        } else if (resultCode == CropImage.CROP_IMAGE_ACTIVITY_RESULT_ERROR_CODE) {
            Exception e = result.getError();
            Toast.makeText( context: this, text: "Possible error is : " + e, Toast.LENGTH_LONG).show();
        }
    }
}
```

Figure 3.6. onActiviteResult function

- The function of choosing the type of the calling card and changing the contact number accordingly.
- A function to verify the selection of the chip, the completion of the image processing, and the extraction of the 15-digit code, then send it.

```
public void selectSim(View v) {
    int RadioButtonId = radioGroupSimCard.getCheckedRadioButtonId();
    if (RadioButtonId == radioButtonMobilis.getId()) { phone = "*111*"; }
    else if (RadioButtonId == radioButtonDjezzy.getId()) { phone = "*700*"; }
    else if (RadioButtonId == radioButtonOoredoo.getId()) { phone = "*222*"; }
}

public void sendSms(View view) {
    body = myText.getText().toString().replace( target: " ", replacement: "");
    AlertDialog.Builder builder = new AlertDialog.Builder( context: this);
    if (phone.length() <= 0) {
        builder.setMessage("Check Sim Sender")
            .setPositiveButton( text: "OK", new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int id) {
                }
            })
            .show();
    } else {
        if (body.length() <= 0) {
            builder.setMessage("Compose 15 figures")
                .setPositiveButton( text: "OK", new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog, int id) {}
                })
                .show();
        } else {
            Intent i = new Intent(Intent.ACTION_CALL);
            try {
                AllCode = phone.concat(body).concat("#");
                Log.d( tag: "MainActivity", msg: "Old AllCode : " + AllCode);
                AllCode = URLEncoder.encode(AllCode, enc: "utf-8");
            } catch (UnsupportedEncodingException e) {
                e.printStackTrace();
            }
            i.setData(Uri.parse("tel:" + AllCode));
            Log.d( tag: "MainActivity", msg: "New AllCode : " + i.getData());
            startActivity(i);
            builder.setMessage(R.string.alert_03)
                .setPositiveButton( text: "OK", new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog, int id) {}
                })
                .show();
        }
    }
}
```

Figure 3.7. sendSms function

iv. Modification to a file *activitymain.xml*

Split the screen horizontally into three sections

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layoutDirection="ltr"
    tools:context=".MainActivity">

    <LinearLayout
        android:layout_width="398dp"
        android:layout_height="549dp"
        android:layout_marginTop="32dp"
        android:gravity="center"
        android:orientation="vertical"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.0">

        <LinearLayout...>

        <LinearLayout...>

        <LinearLayout...>

    </LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>
```

Figure 3.8. file *activity main*

The first section is divided horizontally for the width of the cropped image and the text taken from it to ensure the correctness of the extracted number

```
<ImageView
    android:id="@+id/mImageView"
    android:layout_width="200dp"
    android:layout_height="200dp"
    android:layout_marginTop="50dp"
    android:layout_weight="0"
    app:srcCompat="@mipmap/ic_launcher" />

<TextView
    android:id="@+id/textView"
    android:layout_width="220dp"
    android:layout_height="45dp"
    android:layout_marginTop="30dp"
    android:layout_weight="0"
    android:gravity="center"
    android:layoutDirection="ltr"
    android:text="The code"
    android:textAlignment="center"
    android:textSize="18sp"
    android:textStyle="bold" />

</LinearLayout>
```

Figure 3.9. layout 01 of *activity main*

The second section is divided vertically for the display of the Send the Code button and the Import Image button from the device's applications

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="0"
    android:gravity="clip_horizontal|center"
    android:orientation="horizontal">

    <ImageButton
        android:id="@+id/mButton"
        android:layout_width="142dp"
        android:layout_height="55dp"
        android:layout_marginRight="10dp"
        android:background="#3F51B5"
        android:onClick="onChooseFile"
        android:textAlignment="center"
        android:textColor="#FFFFFF"
        android:textStyle="bold"
        app:srcCompat="@android:drawable/ic_menu_camera" />

    <ImageButton
        android:id="@+id/send"
        android:layout_width="147dp"
        android:layout_height="match_parent"
        android:layout_marginLeft="10dp"
        android:background="#2196F3"
        android:clickable="false"
        android:onClick="sendSms"
        android:textAlignment="center"
        android:textColor="#FFFFFF"
        android:textSize="14sp"
        android:textStyle="bold"
        app:srcCompat="@android:drawable/ic_menu_send" />

</LinearLayout>
```

Figure 3.10. layout 02 of *activity main*

The third section is divided vertically for the width of the radio box set to choose the type of SIM card

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_weight="2"
    android:orientation="horizontal">

    <RadioGroup android:id="@+id/radioGroup_SimCard" android:layout_width="match_parent" android:layout_height="wrap_content"
        android:layout_marginStart="16dp" android:layout_marginLeft="16dp" android:layout_marginTop="24dp" android:layout_marginEnd="16dp"
        android:layout_marginRight="16dp" android:gravity="center" android:orientation="horizontal" app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" app:layout_constraintTop_toBottomOf="@+id/textView3">

        <RadioButton android:id="@+id/radioButton_Mobilis" android:layout_width="wrap_content" android:layout_height="wrap_content"
            android:layout_weight="0" android:background="@drawable/mobilis_radio" android:button="@android:color/transparent"
            android:checked="false" android:onClick="selectSim" android:padding="7dp" android:text="Mobilis" android:textAlignment="center"
            android:textStyle="bold" android:textColor="@drawable/text_mobilis_radio"/>

        <RadioButton android:id="@+id/radioButton_Djezzy" android:layout_width="wrap_content" android:layout_height="wrap_content"
            android:layout_marginStart="10dp" android:layout_marginEnd="10dp" android:layout_weight="0" android:background="@drawable/djezzy_radio"
            android:button="@android:color/transparent" android:onClick="selectSim" android:padding="7dp" android:text="Djezzy"
            android:textAlignment="center" android:textColor="@drawable/text_djezzy_radio" android:textStyle="bold" />

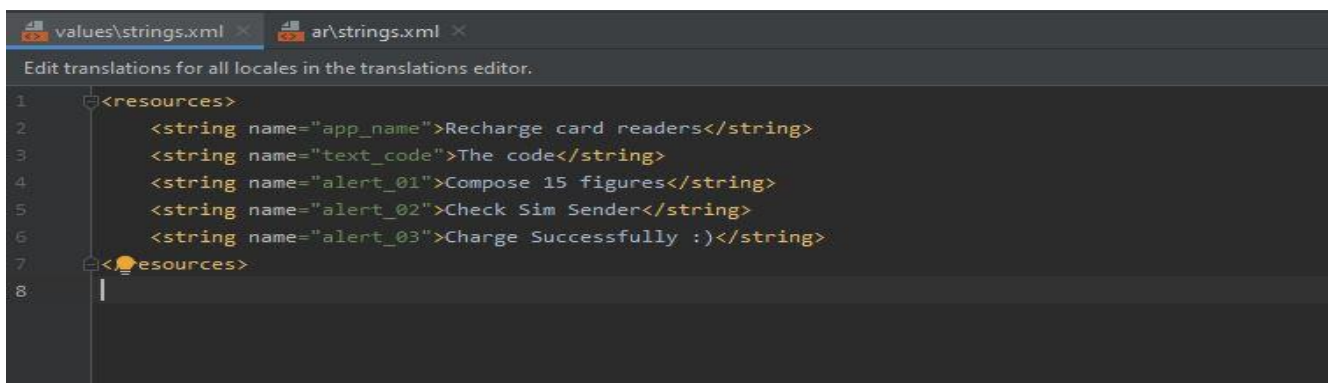
        <RadioButton android:id="@+id/radioButton_Ooredoo" android:layout_width="wrap_content" android:layout_height="wrap_content"
            android:layout_weight="0" android:background="@drawable/ooredoo_radio" android:button="@android:color/transparent"
            android:onClick="selectSim" android:padding="7dp" android:text="Ooredoo" android:textAlignment="center"
            android:textColor="@drawable/text_ooredoo_radio" android:textStyle="bold" />

    </RadioGroup>
</LinearLayout>
```

Figure 3.11. layout 03 of activity main

v. Modification to a files *String.xml*

Add the application name and the phrases for the dialog boxes in the English language and create another file for the Arabic language with the same previous values in the Arabic language in order to appear if the system language is in Arabic



```
values\strings.xml x ar\strings.xml x
Edit translations for all locales in the translations editor.
1 <resources>
2   <string name="app_name">Recharge card readers</string>
3   <string name="text_code">The code</string>
4   <string name="alert_01">Compose 15 figures</string>
5   <string name="alert_02">Check Sim Sender</string>
6   <string name="alert_03">Charge Successfully :)</string>
7 </resources>
8 |
```

Figure 3.12. file string en

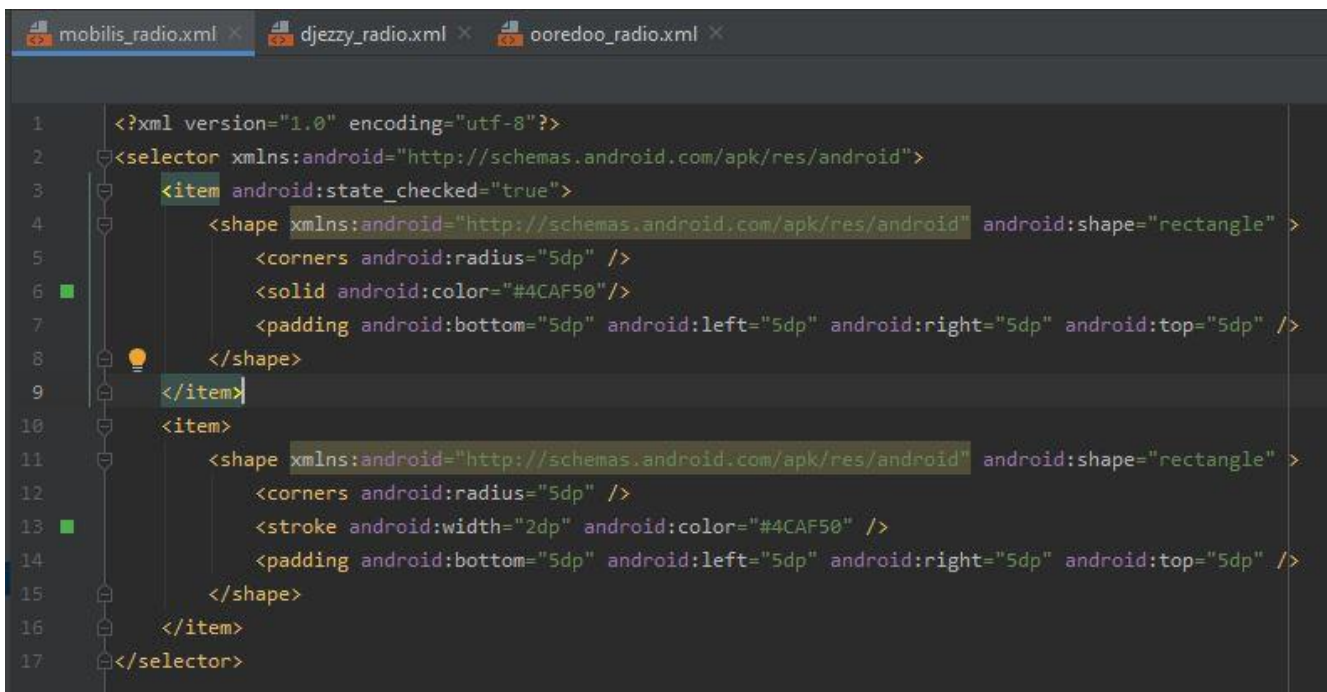


```
values\strings.xml x ar\strings.xml x
Edit translations for all locales in the translations editor.
1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3   <string name="app_name">فراء بطافة سادن الهاتف</string>
4   <string name="text_code">السيطرة</string>
5   <string name="alert_01">ادذل ال 15 رقم</string>
6   <string name="alert_02">ادذل سيطرة ال رقم</string>
7   <string name="alert_03">(: تم السادن بنجاح</string>
8 </resources>
```

Figure 3.13. file string ar

vi. Add properties drawable

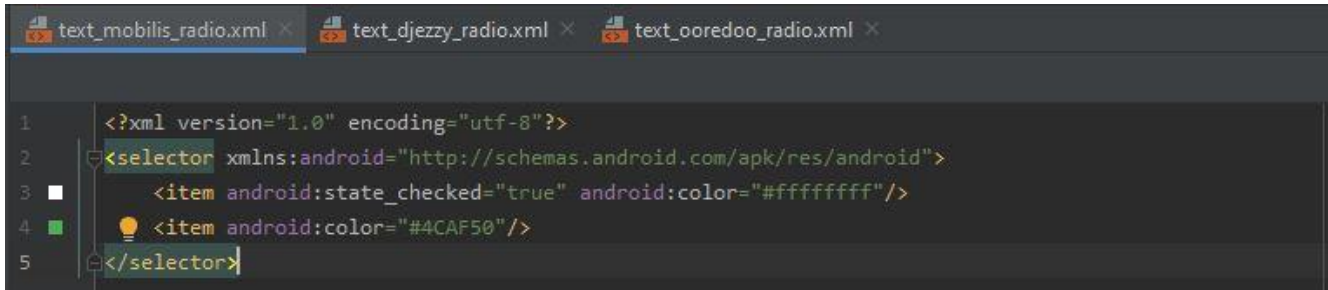
Create three files for changing the shape of the radio box for selecting a sim chip with changing the color in each file



```
mobilis_radio.xml x djezzy_radio.xml x ooredoo_radio.xml x
1 <?xml version="1.0" encoding="utf-8"?>
2 <selector xmlns:android="http://schemas.android.com/apk/res/android">
3   <item android:state_checked="true">
4     <shape xmlns:android="http://schemas.android.com/apk/res/android" android:shape="rectangle" >
5       <corners android:radius="5dp" />
6       <solid android:color="#4CAF50" />
7       <padding android:bottom="5dp" android:left="5dp" android:right="5dp" android:top="5dp" />
8     </shape>
9   </item>
10  <item>
11    <shape xmlns:android="http://schemas.android.com/apk/res/android" android:shape="rectangle" >
12      <corners android:radius="5dp" />
13      <stroke android:width="2dp" android:color="#4CAF50" />
14      <padding android:bottom="5dp" android:left="5dp" android:right="5dp" android:top="5dp" />
15    </shape>
16  </item>
17 </selector>
```

Figure 3.14. file custom background

Create three files for changing the font color inside the radio box upon selection, with changing the color in each file

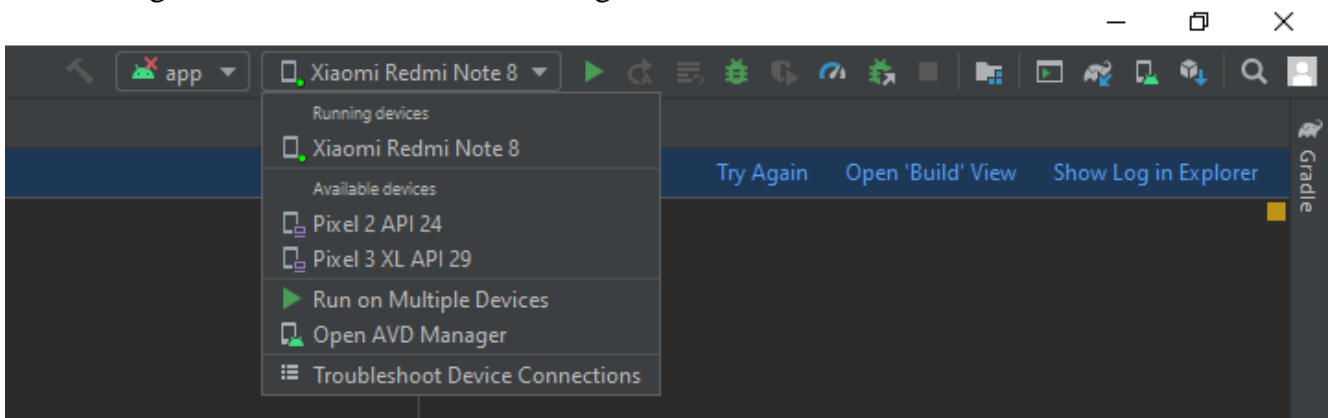


```
1 <?xml version="1.0" encoding="utf-8"?>
2 <selector xmlns:android="http://schemas.android.com/apk/res/android">
3   <item android:state_checked="true" android:color="ffffff"/>
4   <item android:color="#4CAF50"/>
5 </selector>
```

Figure 3.14. file custom text color

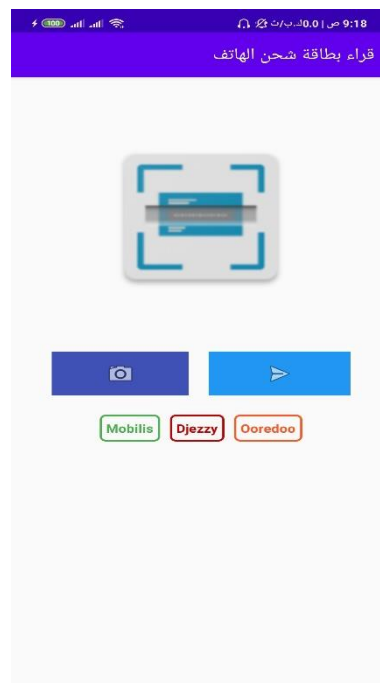
c. Compilation of the project

We integrate the application on the emulator or our phone in order to make sure of the integrity of the design and the code before extracting it in its final form.



4. The results

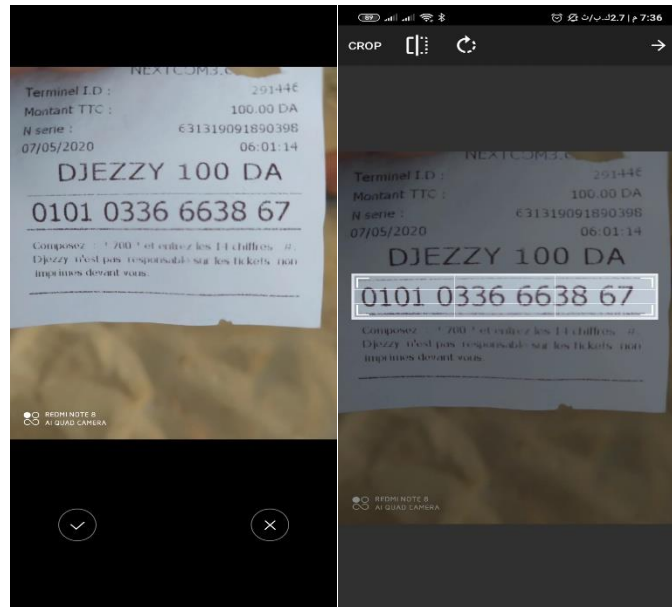
The main interface of the application



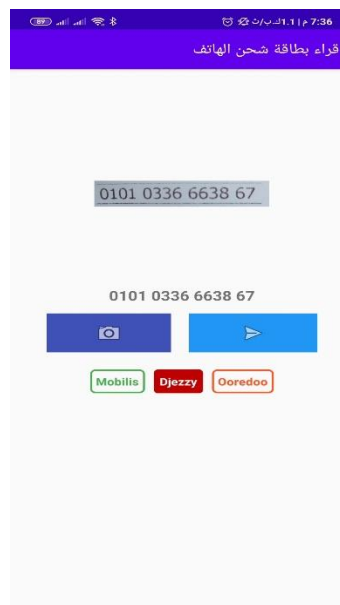
Chapter 3 : Implementation

The application goes through three stages until the number code is entered :

- Import and crop the image :



- Choose the type of chip after extracting the code



- Send the code and complete charging the card

5. Discussion

We encountered two problems in this work:

First, the inaccuracy of the result in the event of confusion next to the numbers, and the most appropriate and simple solution that we saw, which will certainly reduce the appearance of this error, which is to clean the place of the numbers well.

Secondly, the application has not chosen the transmitter chip in the case of more than one chip, and thus this imposes on the user in the case of communication that the sim card is always optional and not a default one.

Of course, there is no perfect job either, with updates and improvements that could be close to that.

V. General conclusion

This project allowed us to get a clearer and more accurate idea about the development on Android, this operating system that has conquered the mobile market.

First of all, we dealt with this system in detail by seeing its history, history and an overview of image processing and the way text is extracted from it.

Then we studied the use of google libraries for image processing and how it works

And our last chapter is related to the design and implementation of our application, which helps the user to enter the code of the phone recharge card without the hassle of entering numbers with his hand, and it was developed with Android Studio.

The main goal has been achieved, but our app is still being improved.

References:

- 01: Link <https://www.geeksforgeeks.org/tesseract-ocr-with-java-with-examples/>
- 02: Link <https://media.geeksforgeeks.org/wp-content/uploads/20190530190230/General-working-of-OCR-3.jpg>
- 03: Link <https://static.googleusercontent.com/media/research.google.com/en/pubs/archive/33418.pdf>
- 04: Link <https://developer.android.com/guide/components/fundamentals>
- 05: Link <https://developer.android.com/guide/topics/manifest/manifest-intro>
- 06: Article Grid Anchor based Image Cropping: A New Benchmark and An Efficient Model. Authors: Hui Zeng, Lida Li, Zisheng Cao, and Lei Zhang
- 07: Website <https://homepages.inf.ed.ac.uk>
- 08: Article An algorithm to Recognize Number in Image: Based on Statistical Image Analysis. Authors: Fathi M. O. Hamed and Amna M. Y. Elnjar
- 09: Link <https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html>
- 10: Website <https://developer.android.com/>

Prospects:

Our work was successfully completed, and we were able to create an android application that enables the user to charge the SIM card easily and effortlessly.

Thus, in the near future, we plan to improve our application, not only adding new functions, but also making the application more interactive with as many slides as possible.